

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PUCP

Estudio de control de posición de un dron DJI Tello con los movimientos

y gestos de la mano

TRABAJO DE INVESTIGACIÓN PARA LA OBTENCIÓN DEL

GRADO DE BACHILLER EN CIENCIAS CON MENCIÓN EN

INGENIERÍA ELECTRÓNICA

AUTOR

Lino Walter Padilla De la Cruz

ASESOR:

Carlos Saito Villanueva

Lima, Noviembre, 2020

Resumen

La evolución tecnológica de los robots ha ido creciendo constantemente durante los últimos años no solo en industrias de producción sino también en la vida social de las personas. Los denominados robots sociales o robots autónomos están ocupando más tareas en la vida diaria de las personas con el pasar de los años. Estos robots están siendo diseñados para servirnos y así tener un estilo de vida más sencillo. Por otro lado, los robots voladores (drones) han sido utilizados con más frecuencia en distintas áreas como la topografía, control fiscal, entre otros. Sin embargo, recientes aplicaciones se están orientando a un uso cotidiano de los drones.

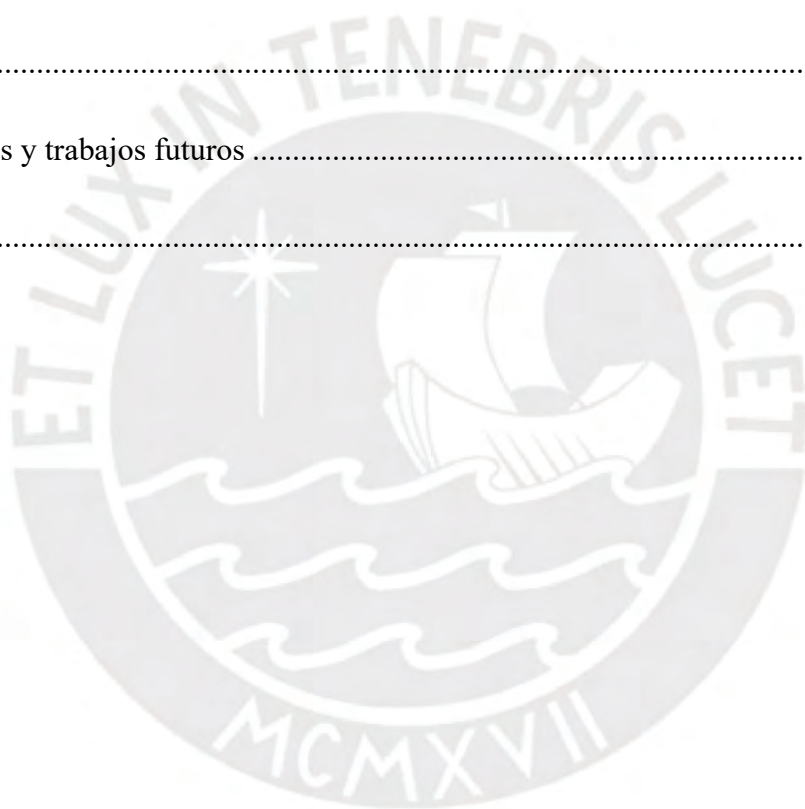
Este objetivo se logrará buscando formas de interacción más fluida con los robots, con el menor uso de dispositivos adicionales. Actualmente los drones son controlados usando un control remoto o joystick lo cual resulta adecuado cuando este se encuentra a una distancia superior a los dos metros. Sin embargo, cuando el dron se encuentra a una distancia menor a los dos metros de las personas se vuelve cada vez más redundante el uso de un joystick para controlar su posición. Diversos estudios demuestran que las personas se muestran más cómodas y seguras controlando al dron utilizando los movimientos y gestos de una mano cuando estos se encuentran cerca.

El presente trabajo de investigación hace una revisión de la literatura basada en el procesamiento de imágenes para reconocimiento de objetos (una mano) y los distintos desarrollos o aplicaciones implementados en drones sociales. La combinación de estas investigaciones busca generar una nueva capacidad de interacción que permita controlar a los drones utilizando los movimientos y gestos de una mano.

Índice General

Capítulo 1: Drones y la interacción con la sociedad.....	1
1.1. Marco problemático	1
1.2. Estado del arte	2
1.2.1. Interacción humano-robot.....	2
1.2.2. Drones.....	3
1.2.3. Dispositivos de detección de gestos.....	6
1.2.4. Lenguajes de programación	8
1.2.5. Estudios realizados para la detección de gestos.....	8
a. Con dispositivo de detección de gestos	8
b. Sin dispositivo de detección de gestos	9
1.3. Justificación.....	11
Capítulo 2: Fundamentos y modelo de solución.....	13
2.1. Marco teórico	13
2.1.1. Diagrama general de solución.....	13
2.1.2. Dispositivos para la solución	14
2.1.3. Teoría sobre procesamiento de imágenes	16
2.2. Marco conceptual.....	17
2.2.1. Preprocesamiento de la imagen	18
2.2.2. Características Haar e imagen integral	18

2.2.3.	Algoritmo AdaBoost y clasificadores en cascada.....	20
2.2.4.	Red neuronal convolucional	24
2.3.	Modelo de solución	26
2.3.1.	Diagrama de bloques.....	26
2.3.2.	Diagrama de flujo	27
2.3.3.	Movimientos y gestos detectados	29
	Conclusiones	31
	Recomendaciones y trabajos futuros	32
	Bibliografía	33



Índice de Figuras

Figura 1 Parrot AR Drone 2.0. [9]	4
Figura 2 Parrot Mambo Fly [10].....	5
Figura 3 DJI Tello [11]	6
Figura 4 Leap Motion Controller [15]	7
Figura 5 Asus Xtion Pro Live Camera [16].....	8
Figura 6 Diagrama de bloques general	13
Figura 7 Modelo de procesamiento de imágenes [28].....	18
Figura 8 Características Haar [29].....	19
Figura 9 Imagen integral [29].....	20
Figura 10 Extracción de características [29].....	21
Figura 11 Algoritmo AdaBoost [30].....	22
Figura 12 Clasificadores en cascada [27]	24
Figura 13 Red neuronal convolucional.....	25
Figura 14 Diagrama de bloques del algoritmo de procesamiento de imágenes.....	26
Figura 15 Diagramas de bloques del dron	27
Figura 16 Diagrama de flujo.....	28

Índice de tablas

Tabla 1. Comparativa de drones	15
Tabla 2. Características de la computadora	16
Tabla 3. Movimientos del dron asociado a la detección de manos.....	29



Introducción

El presente trabajo de investigación está orientado al control de posición del dron DJI Tello usando como medio de comunicación los movimientos y gestos de la mano. Se realiza una investigación previa sobre distintos drones sociales que se pueden emplear con sus respectivas ventajas y desventajas. Además, se analizan los distintos algoritmos u dispositivos que permiten el procesamiento de imágenes para la detección de un objeto en particular (una mano).

En la actualidad existe una tendencia a integrar los robots en la vida diaria de las personas, en especial se observa que los drones son cada vez más accesibles a la comunidad y no solo en industrias. Estos drones son controlados mediante el uso de un control remoto o joystick; sin embargo, estudios recientes demuestran que conforme el dron se vuelve más autónomo se vuelve más redundante usar un dispositivo adicional para controlarlo. Además, estos estudios demuestran que las personas se sienten más cómodos controlando al dron con los movimientos de una mano cuando este se encuentra cerca. Es así, que en el presente trabajo de investigación se analizan las tecnologías para la interacción de las personas y el dron usando como medio de comunicación los movimientos y gestos de una mano para que de esta manera puedan adquirir una nueva capacidad de control.

Este trabajo de investigación puede tener múltiples aplicaciones en el futuro a medida que los drones se vuelvan parte de la vida cotidiana de las personas. Se podrá interactuar con los drones de manera fluida y segura para brindarle órdenes. A personas discapacitadas, por motivos de seguridad, en búsqueda de personas, motivos educativos, entre otros son aplicaciones futuras que se pueden extraer de la presente investigación.

Capítulo 1: Drones y la interacción con la sociedad

En el presente capítulo se menciona el marco problemático que sirve como base para el desarrollo de la presente investigación. También se nombran los distintos estudios y dispositivos usados en desarrollos previos.

1.1.Marco problemático

Los drones o UAV (Unmanned Aerial Vehicle) son, en palabras resumidas, robots voladores que han sido desarrollados y estudiados con mayor frecuencia durante los últimos años por lo que las aplicaciones y uso comercial han ido creciendo. A pesar de que existen aspectos importantes que todavía faltan definir como normas legales de seguridad, no se observa una desaceleración en el uso pues según los datos de la FAA (Federal Aviation Administration) [1] se duplicó la cantidad de drones registrados en los Estados Unidos del 2017 al 2019. Además, se prevé que para el 2022 más de dos millones y medio de drones sean registrados en el país norteamericano con una tendencia al alta.

En un comienzo, los drones realizaban funciones básicas como capturar imágenes y grabar videos con ayuda de la cámara incorporada en su estructura. Sin embargo, debido a nuevos estudios y el avance tecnológico las aplicaciones de los drones se han ampliado. Es así que se puede observar cómo empresas invierten en el uso de drones para servicios militares, cuidados de salud, ayuda humanitaria y alivio de desastres [2]. Por ejemplo, según información de la CB Insights [2] en el 2017 se usaron drones después del huracán Harvey para evaluar daños, localizar víctimas y entregar ayuda.

Las aplicaciones en industrias utilizan principalmente el joystick como dispositivo de control lo cual resulta adecuado en situaciones en la que el dron se encuentre a distancias superiores

a los dos metros. Por otro lado, debido a la creciente popularidad de los drones personales, Sharlin et al [8] menciona que los drones efectuarán un rol social de colaboración en distintas tareas de la vida diaria de las personas. Esto permitirá que los drones se conviertan en una entidad social y que se puedan encontrar a una distancia menor a los dos metros de las personas esperando a recibir una instrucción. Sin embargo, para que los drones sean parte de la sociedad se debe buscar una forma de interacción más fluida y prescindir del uso de un dispositivo de control como el joystick. Estudios de Cauchard [7] y Sharlin [8] demuestran que es más natural usar los movimientos y gestos de la mano para controlar el dron cuando este se encuentra cerca que retirar la mirada hacia el joystick y realizar los comandos. Es así, que el problema que busca analizar este trabajo de investigación es investigar y desarrollar tecnología para la interacción de las personas y el dron usando como medio de comunicación los gestos y movimientos de una mano para que de esta manera se pueda adquirir una nueva capacidad de control.

1.2.Estado del arte

1.2.1. Interacción humano-robot

En 1979 Scheibe y Erwin [3] mencionaron que los humanos tienden a personificar a los robots. También, Reeves y Nass [3] desarrollaron una ecuación que afirma que la interacción entre el humano y el robot es inherentemente social. Además, mencionan que las reglas de la interacción humano-humano también se aplican a la de humano-robot. De esta manera surge el campo de la interacción humano-robot, el cual es una disciplina relativamente nueva que estudia formas de interacción más eficientes y fluidas entre el robot y la persona en aplicaciones domésticas como ayuda, asistentes personales, motivos educativos o como simples artefactos de entretenimiento.

En el estudio realizado por Cauchard et al. [7] se trata de analizar la interacción entre un dron y un humano utilizando como medio de control los movimientos y gestos de una mano. En el estudio se aplica el método Mago de Oz que consiste en hacer creer a un grupo de estudiantes que controlan a un dron la mano mientras un operador escondido realiza los comandos necesarios. Se pudo observar cómo diecinueve estudiantes interactuaban con el dron como si hablaran con una persona o con una mascota y que se sentían más cómodos controlando al dron con la mano en lugar de utilizar un joystick. Esto demuestra que es factible el uso de una nueva forma de controlar el dron cuando este se encuentre cerca.

1.2.2. Drones

Existen dos tipos de drones principalmente: los de ala fija y de ala rotatoria. El primero tiene la capacidad de volar grandes extensiones de terreno lo que permite mapear grandes superficies, por ejemplo, un sembrío en agricultura. El segundo, de ala rotatoria, es el más usado comercialmente pues permite realizar trabajos de inspección, hacer vuelos estacionarios y puede despegar verticalmente desde cualquier superficie. De este último existen diferentes modelos en el mercado.

a. Parrot AR Drone 2.0.

Este dron es de cuatro motores y fue desarrollado por la empresa Parrot. Entre las características principales radica que tiene doce minutos de autonomía, un alcance de control de cincuenta metros y una cámara de 720p a 30 cuadros o fotogramas por segundo. El Parrot AR Drone tiene un sistema operativo Linux que realiza la compresión de imágenes con el formato de codificación H264 y la cámara tiene un ángulo de visión de 92 grados [9].

En los estudios realizados en [4] y [5] se investigó sobre el reconocimiento de gestos de la mano para controlar el movimiento del dron Parrot AR Drone 2.0. con una eficiencia promedio del 90%. Sin embargo, este dron se ha dejado de comercializar por lo que se necesita aplicar el sistema de reconocimiento a un nuevo dron. En la figura 1 [9] se muestra el Parrot AR Drone en la que se puede observar los cuatro motores.



Figura 1 Parrot AR Drone 2.0. [9]

b. Parrot Mambo Fly

Perteneciente a la nueva generación de drones de la empresa Parrot, es un dron cuadricóptero de gran estabilidad que permite ser programado en lenguaje Python y JavaScript, además posee un módulo dedicado en Simulink [8]. También viene con un SDK (Software Development Kit) que permite controlar al dron con comandos simples [10].

Tiene como desventaja que la autonomía es solo de nueve minutos y tiene incorporada una cámara vertical de 60 fotogramas por segundo [10]. Es decir que la cámara se encuentra en la parte inferior por lo que en caso de querer grabar frontalmente se requiere un dispositivo adicional que la empresa Parrot vende. Por tal motivo, no es un dron adecuado en la tarea de

procesamiento de imágenes, prueba de ello es que no existen muchas investigaciones previas con este dron. En la figura 2 [10] se observa el Parrot Mambo Fly sin una cámara frontal.



Figura 2 Parrot Mambo Fly [10]

c. DJI Tello

El dron Tello de la empresa DJI fue diseñado con propósitos educativos. Es un dron de cuatro motores con una autonomía de 13 minutos, puede grabar videos a 720p y tomar fotos de 5 megapíxeles. Además, tiene dos velocidades de vuelo, el lento que es de 10.8 kph y el rápido de 28.8 kph. Como ventaja adicional es que cuenta con diferentes modos de vuelo inteligente: modo rebote, 8D flips, throw & go [11].

Debido al propósito educativo, la empresa DJI desarrolló algoritmos de programación en Python para pruebas de control. Además, existe un foro exclusivo para desarrolladores del Tello y programas de consulta para este dron que permiten mejorar los algoritmos de procesamiento de imágenes.

En la tesis de maestría realizada por Jiménez [12] se utiliza una versión actualizada del Tello, el Tello Edu, en dos etapas principales: el control de un solo dron y el control de múltiples

drones con la detección de cuadrados de colores. Luego, con ayuda de librerías en Python se pudo hacer la conexión entre el dron y la computadora. La figura 3 [11] muestra al DJI Tello.



Figura 3 DJI Tello [11]

1.2.3. Dispositivos de detección de gestos

a. Leap Motion Controller

En la investigación desarrollada por Sarkar et al. [5] se utiliza el Leap Motion Controller, que es un dispositivo de reconocimiento de gestos especializado en el seguimiento de las manos. Este dispositivo tiene un campo de visión (FOV) de 150 grados verticalmente y 120 grados horizontalmente. También posee dos cámaras de 640x240 que tiene una longitud de onda de 850 nanómetros, lo cual está fuera del espectro de luz visible [15]. La precisión del Leap Motion Controller es de 200 cuadros por segundo y puede detectar los movimientos de la mano a una distancia aproximada de 0.9 metros por encima de este [15].

Como desventaja del Leap Motion Controller es que tiene que estar conectado vía USB a una computadora pues este sirve como fuente de energía lo que demuestra una limitante en la portabilidad. También en [5] se demuestra que el dispositivo no detecta la mano cuando se

encuentra en posición vertical y que la eficiencia de la detección se ve mermada al colocar al Leap Motion Controller en ambientes de oscuridad o mucha iluminación. Además, también se detectan errores cuando la cantidad de corriente otorgada por la computadora no es la suficiente.



Figura 4 Leap Motion Controller [15]

b. ASUS Xtion PRO Live Camera

En otra investigación desarrollada por Yangguang Yu et al. [4] se utiliza el dispositivo Asus Xtion Pro Live Camera que brinda una detección no solo de los movimientos y gestos de la mano sino también del cuerpo completo, es decir tiene mayor alcance y efectividad. Usando el Xtion se puede adquirir información de color y profundidad de cada píxel, lo que brinda la capacidad de reconocer hasta nueve gestos y reconocer más de un operador permitiendo el manejo de múltiples drones a la vez [4].

Entre las características principales se encuentra que el FOV es de 58 grados horizontales, 45 grados verticales y 70 grados diagonales. Posee un sensor RGB, de profundidad y un micrófono. Posee una resolución SXGA (1280 x 1024) y detecta las manos a una distancia entre los 0.8m y 3.5m. [16].



Figura 5 Asus Xtion Pro Live Camera [16]

1.2.4. Lenguajes de programación

Existe una vasta cantidad de lenguajes de programación en el mundo tecnológico y entre los más usados se encuentran Python, Java y C++ según un estudio realizado por Hired [17]; por otro lado, en estudios realizados para el procesamiento de imágenes se desarrolló una librería de código abierto denominada OpenCV que posee más de dos mil quinientos algoritmos que permiten realizar reconocimiento de colores, identificar objetos y seguir los movimientos de estos, entre otros [18].

Esta librería tiene compatibilidad con dos de los tres lenguajes más utilizados en la actualidad [17] pero Python resalta entre todos por la sencillez en la sintaxis, tener una curva de aprendizaje corto y poseer otras librerías que complementan el desarrollo de la interacción entre el dron y el humano. A pesar de que Python posee librerías especializadas en el procesamiento de imágenes como Pillow, Skimage, Keras; OpenCV sigue siendo entre todos el más robusto y mejor documentado [14]. Además, resalta la librería Tensorflow con su API Keras para la implementación de redes neuronales convolucionales.

1.2.5. Estudios realizados para la detección de gestos

a. Con dispositivo de detección de gestos

En el estudio realizado por Yu et al. [4] se utiliza como dispositivo de detección de gestos al Asus Xtion para capturar hasta nueve distintos gestos de la mano. También se utiliza en la detección de hasta dos distintos operadores en caso se requiera el control de múltiples drones del modelo AR Drone 2.0.

Debido a que el Asus Xtion puede detectar todo un cuerpo, se elabora un algoritmo adicional que permita a la cámara enfocarse en los gestos de hasta tres operadores. Después estos son interpretados y enviados al Parrot AR Drone y así realice las funciones que se le hayan indicado en la programación. En esta investigación se logra un método accesible con 90% de detección para controlar al dron. El dron responde correctamente a los comandos enviados cuando se encuentra en interiores, pero cuando se realiza el experimento en exteriores la performance disminuye considerablemente.

En otro estudio similar de Sarkar et al. [5] se utiliza el dispositivo Leap Motion Controller que permite detectar cinco posiciones de la mano para mover el dron. En estos dos estudios se utiliza un dispositivo adicional para la detección de gestos que necesitan estar conectados a una estación base o computadora lo que es claramente una limitante en la portabilidad.

b. Sin dispositivo de detección de gestos

Un estudio que deja de utilizar los dispositivos de detección de gestos es realizado por Nguyen et al. [6] pues se emplea la cámara del dron para analizar si lo que se está observando es una mano y si está realizando un gesto. Esto lleva a quitar la limitante de la portabilidad. Esta investigación [6] hace uso de un algoritmo de aprendizaje automático aplicando dos conceptos conocidos como “Haar feature” y “AdaBoost classifier” que fueron propuestas en

los noventa pero que en los últimos años han sido utilizadas frecuentemente debido al avance de la inteligencia artificial.

En [6], la imagen que se captura ingresa a un clasificador que se forma haciendo uso de los conceptos de imagen integral, características Haar, clasificador AdaBoost y clasificador en cascada y obtener como resultado si se detecta un gesto u otro. El algoritmo aprendizaje automático opera satisfactoriamente debido a que en la etapa de entrenamiento se le proporciona un conjunto de 8302 imágenes en las que se varía la distancia del gesto de la mano, los fondos y la iluminación. Finalmente, se logra tener una eficiencia promedio 90% en ambientes iluminados, con un fondo blanco y siempre que la mano se encuentre a una distancia no mayor a 1 metro.

Las redes neuronales convolucionales, por otra parte, son una aplicación moderna de aprendizaje automático que permite reemplazar la característica Haar y al clasificador AdaBoost. Los estudios realizados en [22], [23] y [24] aplican los conceptos de redes neuronales convolucionales para detectar, en un caso, gestos de números en la mano y, en otro caso, detectar la lengua de signos americana (ASL).

El estudio realizado por Dhall et al. [22] construye una red neuronal de siete capas ocultas con el objetivo de detectar los gestos de la mano que indican un número del 1 al 5 logrando una eficiencia del 99.13%. Este objetivo se logra con la obtención de un dataset que contenga imágenes de los gestos de la mano variando las condiciones de iluminación y distancia. Por otro lado, el estudio realizado por Islam et al. [23] entrena una red neuronal con 9360 imágenes con los gestos del ASL logrando una eficiencia de reconocimiento del 94.57%. De esta manera se muestra un avance significativo para las personas con discapacidades (sordomudas) pues se podrán comunicar e interrelacionar con los robots. Sin embargo, estos

reconocimientos no se realizan en tiempo real sino se le entregan imágenes una por una para que el algoritmo lo detecte.

1.3. Justificación

A pesar de que el campo del HRI es relativamente nuevo, se puede notar que los robots están siendo empleados con mayor frecuencia en ámbitos cotidianos. En particular el HDI (Interacción humano dron) tiene como objetivo conseguir que los drones puedan pertenecer a la vida diaria de los humanos y ayudar que las tareas de las personas sean más sencillas. Como se indica, Sharlin et al. [8] menciona que cuando un dron se encuentra cerca de un operador es más natural controlar el dron con los gestos que apartar la mirada y realizar los comandos con un joystick. Por otra parte, Cauchard et al. [7] menciona que mientras más autónomo se vuelve el dron más redundante se vuelve el uso del control remoto.

Cauchard [7] indica que una funcionalidad que se le pueda brindar al dron y en la que se está trabajando es la de que te siga y proporcione información sobre tu estado mientras haces ejercicio. En caso de personas sordomudas y con dificultades físicas, que el dron tenga la capacidad de interpretar tus gestos y pueda realizar una tarea de acuerdo a lo que visualiza resultaría en un gran avance tecnológico. En situaciones de búsqueda y rescate, si el dron detecta los gestos y movimientos de personas atrapadas en escombros o de una persona ahogándose y realice una acción sería una diferencia clave para salvar la vida humana. Estas situaciones requieren una tasa alta de detección de gestos. De tal manera que el presente trabajo de investigación se muestran los algoritmos de procesamiento de imágenes para la detección de una mano que permita al dron desplazarse y realizar acciones.

Por los motivos expresados, se observa que existe una nueva oportunidad de investigar y desarrollar una nueva tecnología que nos permita comunicarnos con el dron de manera fluida y natural. Que pueda ser implementado en áreas educativas y en la vida cotidiana de las personas. De esta manera se abre las posibilidades a futuras investigaciones para desarrollar e implementar el presente trabajo de investigación



Capítulo 2: Fundamentos y modelo de solución

En el presente capítulo se presenta el marco teórico, marco conceptual como conocimiento base para una futura implementación. Además, se menciona el modelo de solución y metodología que se implementa en estudios previos.

2.1.Marco teórico

En esta sección del capítulo se presenta un diagrama de bloques general, los dispositivos que se usan y los estudios en el cual se basa el trabajo de investigación.

2.1.1. Diagrama general de solución

El diagrama de bloques general es el siguiente:

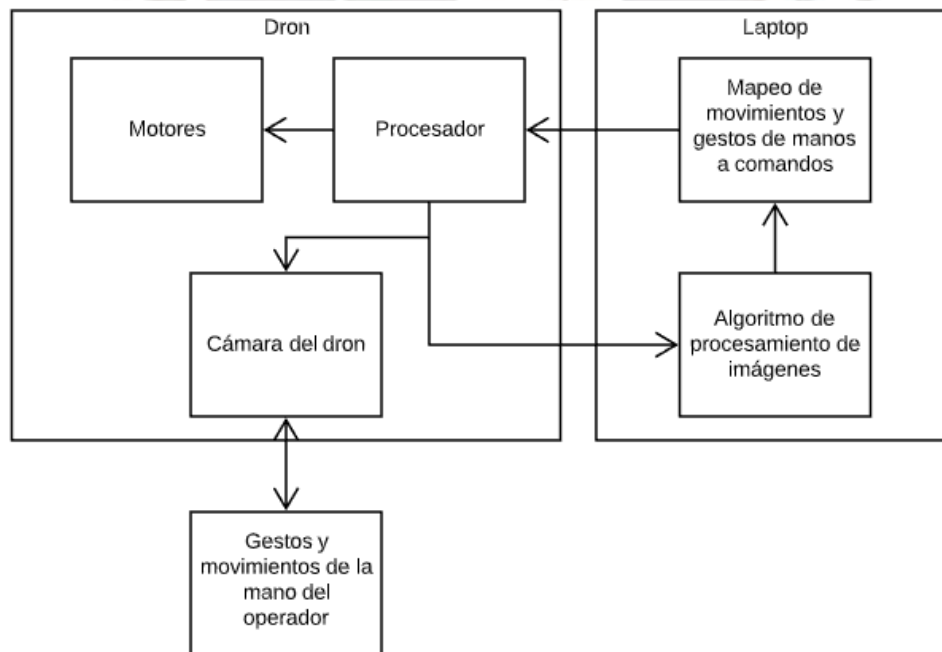


Figura 6 Diagrama de bloques general

El diagrama general de solución muestra que existen dos componentes principales: el dron y la computadora o laptop. Ambos dispositivos intercambian información inalámbricamente usando la tecnología Wifi. Por otro lado, la cámara del dron captura constantemente los movimientos y gestos de la mano que realiza el operador.

El dron o UAV consta de tres partes principales:

- Cámara: Es el componente o sensor que captura el entorno en tiempo real.
- Procesador: Es la parte central del dron que permite la comunicación con la computadora y manipula los motores.
- Motores: Son los accionadores de vuelo del dron.

En la computadora se implementa:

- Algoritmo de procesamiento de imágenes: Es la lógica que se almacena en la computadora usando el lenguaje Python y que implica realizar un modelo de aprendizaje automático para el reconocimiento de la mano.
- Mapeo de movimientos y gestos de manos a comandos: Es un algoritmo que también se almacena en la memoria de la computadora y que permite asociar un movimiento o gesto a una acción que debe realizar el dron.

2.1.2. Dispositivos para la solución

En la selección del dron se escogió el DJI Tello dentro del mercado por las características que ofrece en comparación con el Parrot Mambo Fly. Ambos son drones pequeños y en la tabla 1 se puede observar una comparativa entre estos dos modelos pertenecientes a diferentes empresas.

	DJI Tello	Parrot Mambo Fly
Autonomía	13 minutos	9 minutos
Rango	100 metros	100 metros
Cámara	5MP	0.9MP
Video	720p	720p
Estabilización de imagen	Electrónico	No
Costo	\$125	\$100
Conectividad	Wifi	Wifi

Tabla 1. Comparativa de drones

Estos dos drones ofrecen características similares; sin embargo, se puede observar que el DJI Tello ofrece tres características importantes que lo diferencian del Parrot Mambo Fly. La primera es la autonomía de vuelo pues el Tello puede estar en vuelo controlado más tiempo que el Mambo. La segunda es la calidad de imagen y la tercera es que el Tello tiene estabilización de imagen electrónica. Por otro lado, en muchos estudios realizados con drones se utiliza el Parrot AR Drone con el cual se obtienen buenos resultados. Sin embargo, este dron ya no se encuentra disponible en el mercado por lo que se desarrolla con el DJI Tello por dos razones: las ventajas que ofrece con respecto al Parrot Mambo Fly mencionadas previamente y que el Parrot AR Drone ya no se comercializa.

En cuanto a la elección del procesador de imagen, se puede encontrar diferentes alternativas como el Raspberry Pi 4 que tiene la capacidad de reproducir videos 4Ka 60fps, pero debido a que se busca una interacción más fluida entre el dron y el operador se opta por utilizar la menor cantidad de dispositivos adicionales. De esta forma se opta por usar la computadora

como el único dispositivo de procesamiento de imágenes capturadas por la cámara del dron. A continuación, se muestran las características principales de la computadora Lenovo que se utiliza.

Característica	Valor
Procesador	Core i5-4200U (1.6GHz a 2.6GHZ)
Almacenamiento	1TB
Memoria RAM	8GB
Sistema operativo	Windows 10
Puertos	2 USB 3.0., 1USB 2.0., 1 HDMI
Conectividad	Wifi 802.11 b/g/n

Tabla 2. Características de la computadora

2.1.3. Teoría sobre procesamiento de imágenes

La imagen digital se puede definir como una función matemática de dos dimensiones $f(x, y)$ en la cual x e y son coordenadas espaciales cuya referencia comienza en la parte superior izquierda de cada imagen. A cada posición le corresponde un valor que en escala de grises representa la intensidad de luz en ese punto. De esta forma se observa que en este formato solo se necesita una matriz de dos dimensiones para representar la imagen. Por otro lado, si se tiene una imagen a color RGB se necesita una matriz de tres dimensiones para representar la intensidad de cada color (rojo, verde, azul).

En la etapa de procesamiento se utilizará desenfoque gaussiano o filtro de desenfoque que es una operación que permite eliminar ruidos de alta frecuencia en la imagen que se recibe, es

decir píxeles cuyo valor cambia rápidamente. Se aplica la siguiente fórmula gaussiana a cada píxel de la imagen, donde σ representa la desviación estándar de una distribución gaussiana. Con esta función se construye una matriz de convolución que es aplicada a la imagen original.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Luego de aplicarse el filtro gaussiano es necesario la conversión a escala de grises de la imagen recibida utilizando una suma ponderada. A cada intensidad de color se le aplica un peso distinto con el objetivo de obtener un único valor de intensidad.

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

El algoritmo debe enfocarse en la imagen de la mano, para ello se aplica el método de umbralización binaria. Esto se aplica a cada píxel con la siguiente fórmula estableciendo primero un valor umbral. Si el valor de intensidad del píxel es menor al valor umbral se le da el valor 0 (negro) y si es superior al valor umbral se le da el valor de 1 (blanco). De esta forma se obtiene una imagen binaria.

$$f(x, y) = \begin{cases} 1, & f(x, y) > T \\ 0, & f(x, y) \leq T \end{cases}$$

2.2.Marco conceptual

Esta parte del trabajo de investigación presenta los conceptos necesarios que permita un entendimiento correcto de la solución propuesta. El modelo de procesamiento de imágenes sigue el siguiente proceso. Los datos de entrada son las imágenes o video en tiempo real, luego se realiza el preprocesamiento de las imágenes que ayudan a tener un tiempo de

ejecución del algoritmo más rápido. De estas imágenes se extraen las características más importantes que nos ayudan a detectar una mano y finalmente se obtiene como respuesta la probabilidad de que la imagen contenga efectivamente una mano.

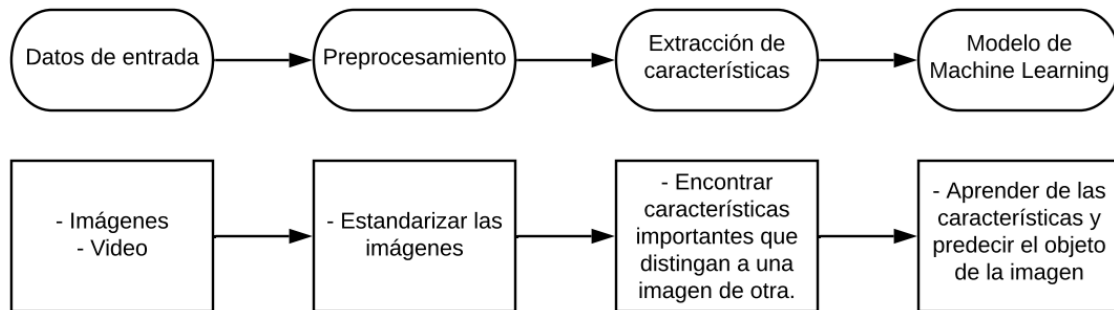


Figura 7 Modelo de procesamiento de imágenes [28]

2.2.1. Preprocesamiento de la imagen

Las imágenes serán capturadas por la cámara del dron y enviadas a la computadora para que pueda ser procesada. Estas imágenes entran a una etapa de preprocesamiento o estandarización que implica, por ejemplo, convertir las imágenes RGB a una escala de grises, esto con el objetivo de que el procesamiento posterior sea más rápido pues la representación de una imagen a color necesita una matriz de tres dimensiones mientras que en escala de grises se necesita una matriz de dos dimensiones lo que permite un menor uso de memoria.

2.2.2. Características Haar e imagen integral

La característica Haar son unos pequeños cuadrados o rectángulos de color blanco y negro que permiten extraer información importante de una imagen como bordes, líneas rectas y diagonales que ayudan, en última instancia, a reconocer un objeto. En la figura 8 se muestran las cuatro características fueron utilizadas por Viola-Jones [25]; sin embargo, Nguyen et al.

[6] utiliza otros adicionales que permitan la detección de los gestos de una mano. También en el estudio de Yun et al. [26] se utilizan unas características adicionales para la detección de tres posturas de la mano pues los usados por Viola-Jones no son suficientes para una buena tasa de detección. De esta forma para detectar la palma de una mano se tiene una tasa de detección del 97.5% usando una extensión de las características Haar y una detección del 93.3% usando solamente las cuatro características Haar mostradas en la figura 8.

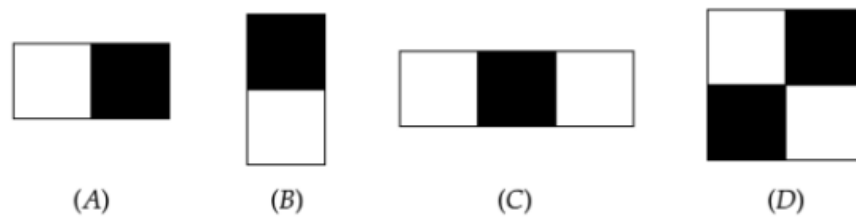


Figura 8 Características Haar [29]

A cada característica Haar se le asocia un valor que se calcula sumando los píxeles que se encuentran en la parte oscura y restándole la suma de los píxeles que se encuentran en la parte blanca. Esta característica se va desplazando por toda la imagen buscando un valor que se aproxime a un valor establecido previamente o supere un límite umbral, de esta forma si un valor encontrado supera el límite umbral se indica que esta característica Haar es un buen indicador para encontrar el objeto deseado. Las características Haar pueden tener distintos tamaños por lo que la cantidad de píxeles en el interior puede ser grande. Esto conlleva a que calcular el valor de una sola característica representa un gasto computacional elevado ya que se necesita recorrer toda la imagen constantemente y sumar los píxeles en la región en cada iteración. Por eso Viola-Jones establecen una representación intermedia de la imagen de entrada que se denomina imagen integral.

El cálculo de imagen integral en una posición (x, y) se realiza sumando todos los píxeles que se encuentran en la parte superior, en la parte izquierda del píxel (x, y) y este inclusive. Esta operación matemática se realiza en toda la imagen en escala de grises. De esta forma si se quiere calcular la suma de píxeles en la posición D de la figura 9 [29] solo bastaría con sumar la imagen integral de 1 y 4 y sustraer la suma de 2 y 3. Así, si se desea calcular el valor de una característica Haar de dos rectángulos adyacentes se necesita operar seis valores; para tres rectángulos, ocho valores y para cuatro rectángulos, nueve valores independientemente del tamaño de la característica Haar.

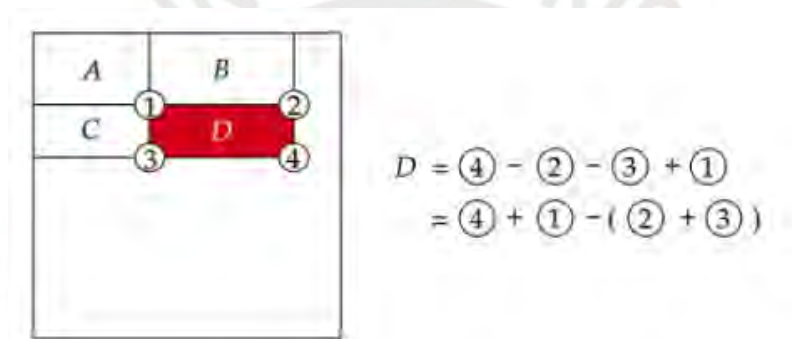


Figura 9 Imagen integral [29]

2.2.3. Algoritmo AdaBoost y clasificadores en cascada

El algoritmo de Viola-Jones utilizan subventanas de 24×24 píxeles mientras que Nguyen et al. [6] utiliza subventanas de 20×20 pues según una investigación de Lienhart se alcanzan mejores tasas de detección con este tamaño de cuadro. Estas subventanas se van desplazando por toda la imagen de entrada analizando si encuentran el objeto deseado. A pesar de que la representación en imagen integral reduce considerablemente el tiempo de cálculo de las características Haar, se puede observar que dentro de la subventana se llega a calcular cerca

de ciento sesenta mil características de las cuales no todas son importantes para la detección de un objeto en particular.

El algoritmo de aprendizaje automático AdaBoost permite obtener las características más importantes o los que permitan reconocer una mano y de entrenar al clasificador. En la figura 10 [29] se tiene n características Haar que como se menciona son cerca de ciento sesenta mil en una subventana. El algoritmo AdaBoost es un procedimiento que se realiza T veces, siendo T el número de las mejores características que se quieren obtener. En el estudio de Viola-Jones se extraen 6061 características que ahora se denominan clasificadores débiles. En un primer instante el algoritmo AdaBoost les otorga el mismo peso a todos, pero los va actualizando conforme se va evaluando nuevas imágenes y se determina que característica ayuda mejor en la detección de una mano. Posteriormente se realiza una combinación lineal de los clasificadores débiles (características Haar) y se forma un clasificador fuerte que permite determinar si la imagen de entrada contiene una mano.

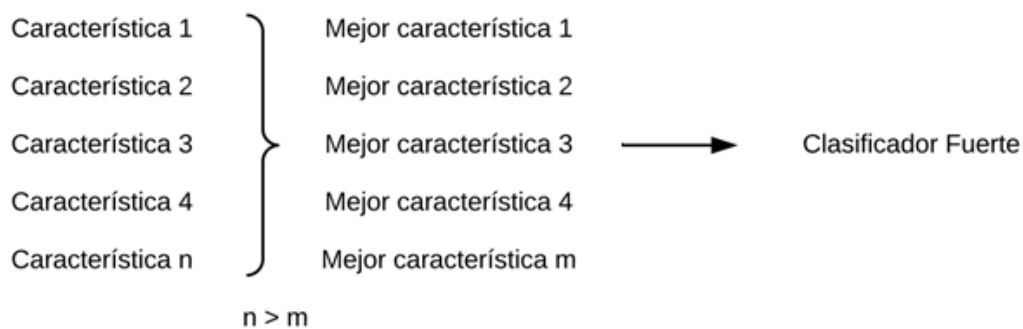


Figura 10 Extracción de características [29]

En la figura 11 [30], se tiene como objetivo separar los signos positivos (mano) de los negativos (no mano). El algoritmo de AdaBoost otorga los mismos pesos a todos los valores

del conjunto de datos de entrada y realiza una primera clasificación que trata de separar los signos positivos y negativos. Como se observa en la primera iteración se obtiene tres errores (dos negativos se encuentran en la zona de positivos y un positivo en la zona de negativos) por lo que a estos valores errados se le otorga un mayor peso en la siguiente prueba para que el algoritmo se enfoque en clasificar estos errores. En la siguiente iteración, el clasificador logra separar los errores en el grupo correspondiente, pero tiene nuevos errores en clasificar otros signos. De esta manera a cada signo se le va actualizando los pesos en cada iteración. Finalmente, se observa que se tiene tres iteraciones, tres clasificadores débiles y con ellos se forma un clasificador fuerte que permite separar con mayor efectividad ambos grupos de signos.

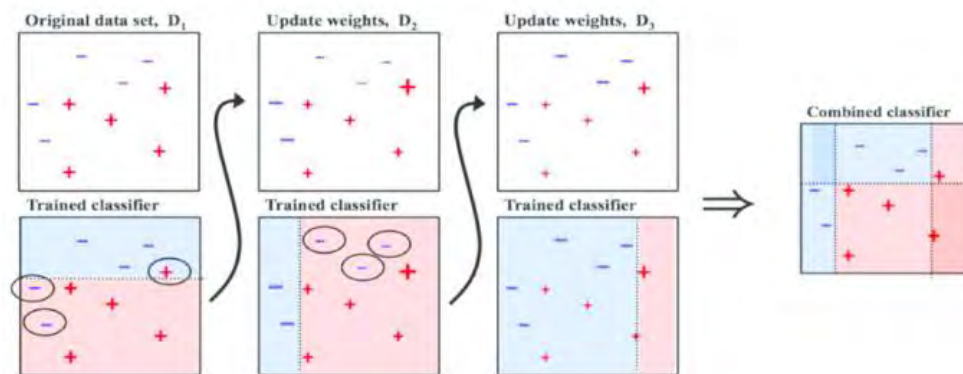


Figura 11 Algoritmo AdaBoost [30]

En busca de reducir aún más el tiempo de procesamiento se emplea los clasificadores en cascada que es un conjunto de clasificadores fuertes distribuidos secuencialmente. El algoritmo de Viola-Jones [25] construye un clasificador en cascada de 38 estados en la que cada estado es un clasificador fuerte. Cada clasificador fuerte tiene diferentes cantidades de

clasificadores débiles. En el primer estado se coloca el clasificador con mayor eficiencia que permite detectar una cara eliminando una gran cantidad de subventanas que no contienen una cara reduciendo el tiempo de procesamiento, para ello se le coloca un umbral mínimo y mantener los falsos negativos casi cero, esto conlleva a tener una cantidad alta de falsos positivos (alta tasa de detección). El diseño del clasificador en cascada debe escoger el número de estados (clasificador fuerte) que se quiere tener, número de características en cada estado y un umbral en cada clasificador. En un busca de obtener un clasificador óptimo se debe establecer la tasa máxima de falsos positivos por estado (f_i), la tasa mínima de positivos verdaderos por estado (d_i) y la tasa de falsos positivos de todo el clasificador (F_i). De esta forma hasta que no se alcance el F_i se debe seguir agregando estados (clasificadores fuertes) y hasta que no se alcance las tasas f_i y d_i se sigue agregando nuevas características al clasificador fuerte en ese estado. Se debe considerar que mientras el número de estados aumenta, el número de “no manos” categorizados como “manos” decrece, pero aumenta el tiempo de procesamiento.

En el algoritmo de Nguyen et al. [6] se detecta cinco gestos de una mano y se entrena un clasificador para cada uno que se entrenan de manera paralela obteniendo 4 estados para el primer clasificador, 16 para el segundo, 8 para el tercero, 10 para el cuarto y 5 para el último. Igualmente, en el estudio de Yun et al. [26] se entrenan tres clasificadores en paralelo y en el estudio de Chen et al. [27] se utiliza cuatro clasificadores en paralelo.

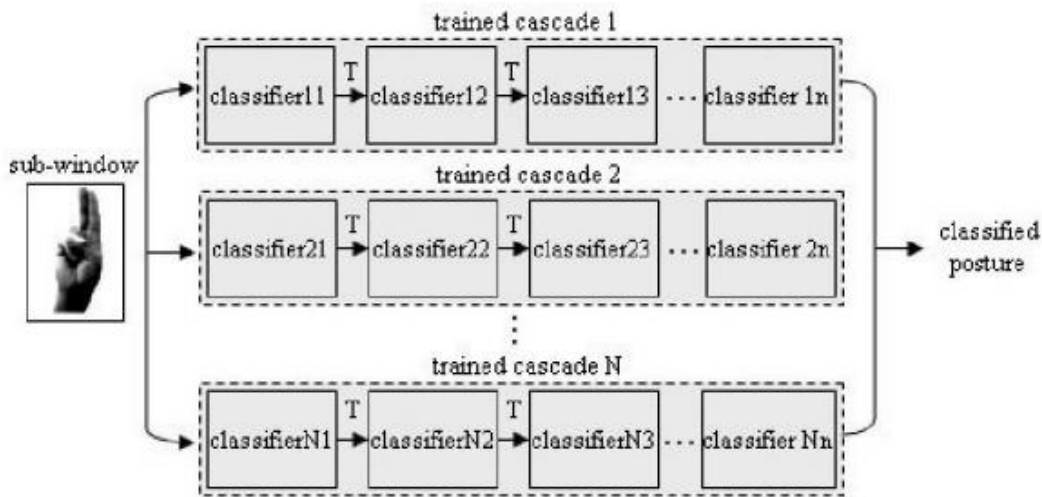


Figura 12 Clasificadores en cascada [27]

2.2.4. Red neuronal convolucional

La red neuronal convolucional (CNN) es una red artificial perteneciente al grupo de aprendizaje supervisado compuesto por varias capas ocultas con diferentes cantidades de neuronas. Cada capa oculta se encarga de capturar e identificar distintos aspectos de una imagen de entrada; conforme se va ingresando a capas más profundas la red puede detectar características más complejas. Este algoritmo permite reemplazar a las características Haar y a los clasificadores en cascada AdaBoost en un solo bloque de neuronas con múltiples capas. Además, al diseñar la red eficientemente ya no se necesita calcular valores manualmente como se realiza en el algoritmo de Viola Jones en virtud de que los cálculos de pesos de los clasificadores lo realizan la red neuronal.

Este modelo de aprendizaje automático aprende por sí sola por lo que en un primer momento se necesita entrenar la red con una vasta cantidad de imágenes (más de quinientos por cada clase). Las imágenes deben contener la clase a ser detectada en distintas posiciones y

condiciones para que la red pueda reconocer variaciones de la clase. Además, es importante tener en cuenta la dimensión y el color de la imagen debido a que esto determina el número de neuronas en la capa de entrada. Es decir que tener una imagen de 12x12 píxeles implica tener una capa de entrada con 144 neuronas si la imagen se encuentra en escala de grises, pero 432 neuronas si la imagen se encuentra en RGB.

Luego se aplican las convoluciones que consiste en realizar una multiplicación escalar de grupos cercanos de píxeles con un kernel que es similar a la aplicación de una característica Haar. Este kernel recorre toda la imagen generando una nueva matriz que será la primera capa oculta. Después de esto se aplica un filtro que es un conjunto de kernel (alrededor de 32 o 64) dependiendo de la estructura de la red neuronal obteniendo en la salida un conjunto de 4608 neuronas en el caso de utilizar un filtro de 32 kernel. Finalmente se aplica el subsampling (Max-Pooling) que permite reducir el número de neuronas en la cual se contengan solo las características más importantes. En el caso de una imagen de 12x12 al aplicar un Max-Pooling de 2x2 se obtendría una imagen de 6x6. Al finalizar todo este proceso se vuelve a aplicar una nueva convolución.

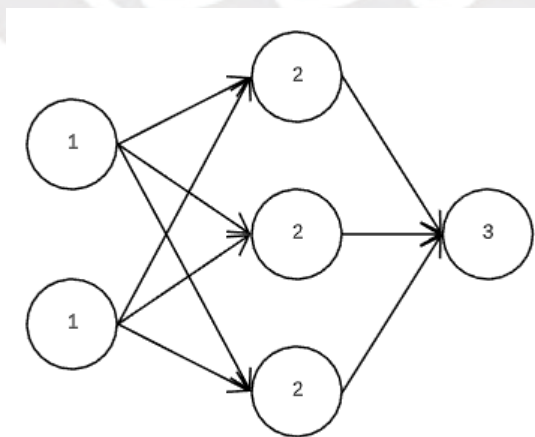


Figura 13 Red neuronal convolucional

2.3. Modelo de solución

2.3.1. Diagrama de bloques

A continuación, se muestra el diagrama de bloques de la computadora que es el dispositivo donde se realiza los códigos en Python y se aplica los conceptos mencionados previamente.

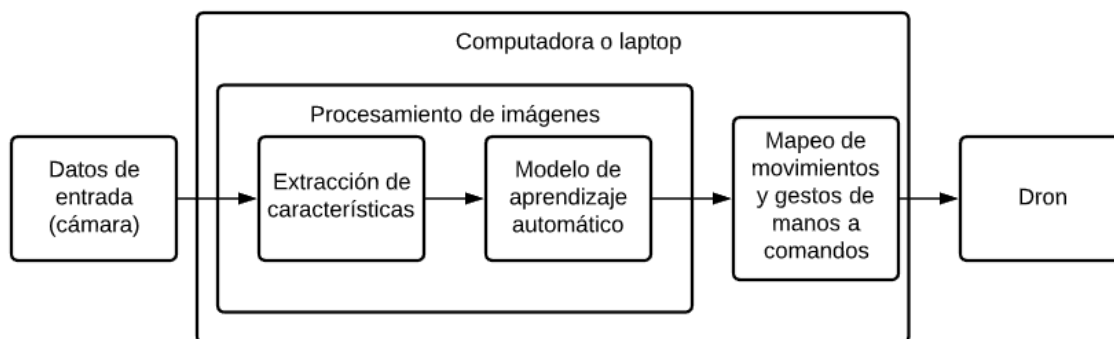


Figura 14 Diagrama de bloques del algoritmo de procesamiento de imágenes

En la imagen se observa que la computadora recibe las imágenes capturadas por el dron utilizando el Wifi. Luego, estas imágenes son preprocesadas para que se encuentren en un formato único y así el algoritmo de procesamiento de imágenes realice un cálculo más rápido. El primer algoritmo a utilizar será los clasificadores en cascada Adaboost que realiza la captura de las características más importantes que nos permitan identificar una mano. Estas características se almacenan en un vector para que el modelo de aprendizaje automático obtenga las mejores características y forme un clasificador fuerte que permita la detección. El segundo algoritmo a utilizar es una red neuronal ya construida denominada YOLOv4 debido a su eficiencia y rapidez. Finalmente, el algoritmo de mapeo de movimientos y gestos de mano a comandos permite detectar un gesto de la mano detectada y asociarlo a una acción en particular que deba realizar el dron. Estos modelos se pueden mejorar si se realizan otros

métodos de preprocesamiento de las imágenes, se cargan más imágenes de entrenamiento o extrayendo mejores características de las entradas.

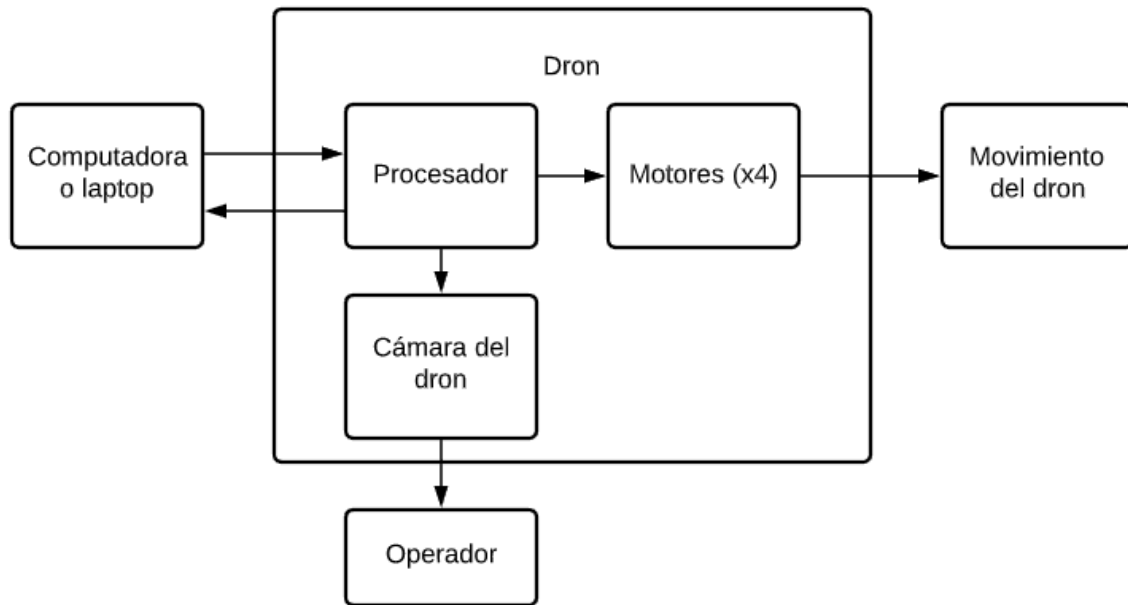


Figura 15 Diagramas de bloques del dron

El diagrama de bloques del dron en la figura 14 muestra que los gestos mapeados a un comando se envían específicamente al procesador del dron que interpreta estos comandos y envía la señal a los cuatro motores para que el dron se mueva a la posición deseada. Además, el procesador envía a la computadora las imágenes que se captura con la cámara.

2.3.2. Diagrama de flujo

El diagrama de flujo muestra la lógica general que se emplea en el reconocimiento de la mano y la acción que realizará el dron. El proceso de captura de la imagen y video es iterativo pues constantemente se está intercambiando información entre el dron y la computadora. Luego se detecta una mano y en el caso de ser solo una mano realiza el seguimiento de esta y analiza

si se detecta el gesto de tres dedos levantados para tomar foto. Caso contrario (detecta dos manos dos manos) el dron realiza un vuelo estacionario y espera a realizar una acción en particular de acuerdo al movimiento realizado por las dos manos como un flip a la izquierda, flip a la derecha o una rotación horaria. Si no detecta mano el dron solo realiza vuelo estacionario.

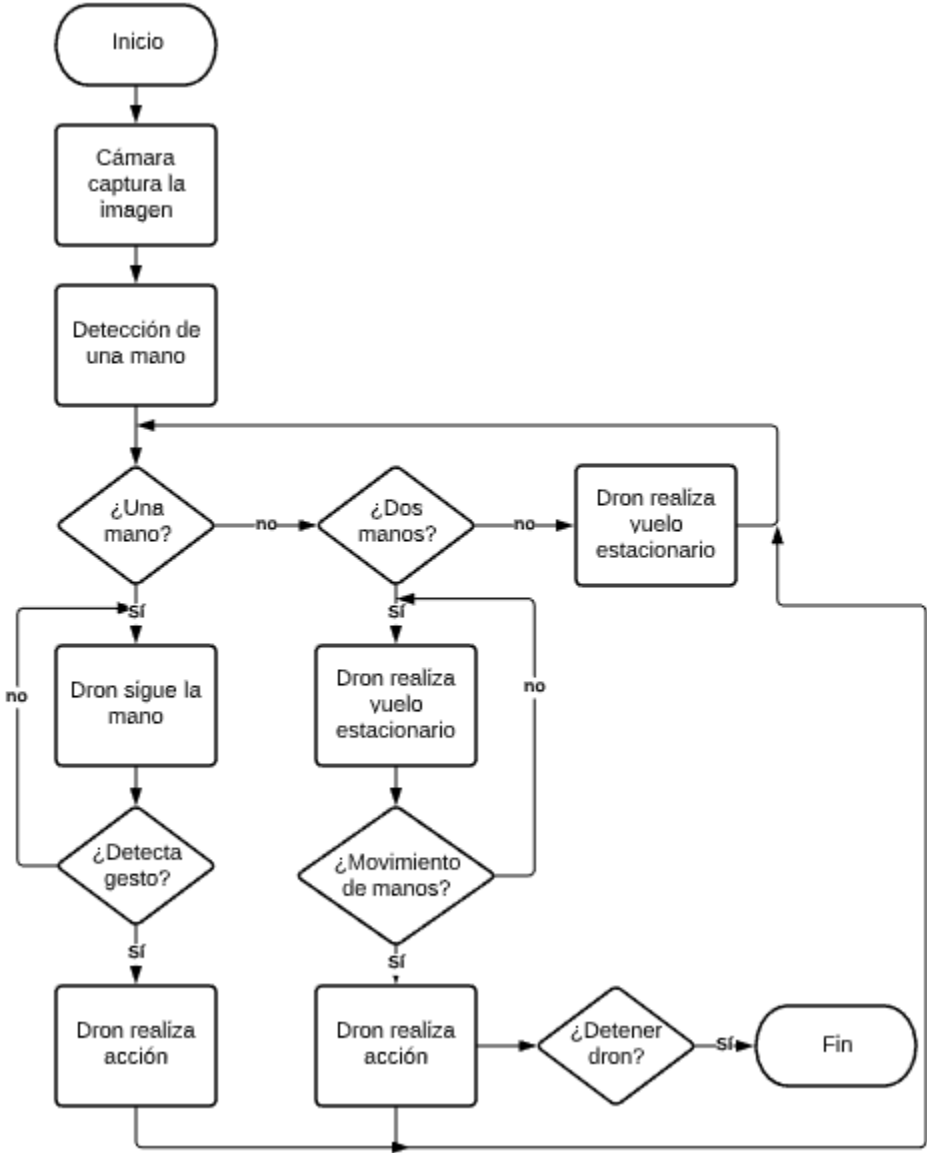


Figura 16 Diagrama de flujo

2.3.3. Movimientos y gestos detectados

Existen distintos algoritmos entrenados para los clasificadores AdaBoost que se encuentran disponibles en GitHub en formato xml y que son de libre disponibilidad. En especial se encuentran disponibles archivos asociados a detección de una mano ya entrenada que permite la futura implementación del presente trabajo de investigación. La detección y movimientos se realizan cuando el dron se encuentra a distancias menores a los dos metros de distancia pues a distancias mayores la resolución de la cámara no permite una buena detección. En la tabla 3 se muestran las acciones que realiza el dron de acuerdo a la detección de las manos.

Cantidad de manos	Acción del dron	Movimientos de manos	Piruetas del dron
1	Seguimiento	Gesto (tres dedos)	Toma foto
2	Parada	A la derecha	Flip izquierda
2	Parada	A la izquierda	Flip derecha
2	Parada	Hacia arriba	Flip atrás
2	Parada	Hacia abajo	Flip adelante
2	Parada	Izquierda cerca	Rotación horaria
2	Parada	Derecha cerca	Rotación antihoraria

Tabla 3. Movimientos del dron asociado a la detección de manos

Los dos algoritmos para la detección de una mano que deben implementarse: clasificador en cascada AdaBoost y la red neuronal convolucional. Se debe analizar en cada uno el tiempo que le toma al algoritmo procesar una imagen para la detección (tiempo de procesamiento),

la exactitud en la detección de una mano variando la distancia, condiciones de iluminación y los fondos de las imágenes.



Conclusiones

Se puede concluir que se debe desarrollar la capacidad interacción humano – robot (en particular humano – dron) utilizando los movimientos y gestos de la mano para controlar los movimientos de un dron DJI Tello.

Esto se logra dividiendo la presente investigación en cuatro tareas:

El primero consiste en diseñar el concepto de solución planteada en el presente trabajo de investigación utilizando las herramientas necesarias para aplicar los conceptos de visión por computadora.

Segundo, realizar la comunicación del dron a una laptop o estación base para el envío de comandos básicos que permitan el movimiento del dron. Además, esta comunicación debe permitir el envío de imágenes y así tener la capacidad de procesarlas y realizar transformaciones.

Tercero, desarrollar la capacidad de visión por computadora para detectar una mano, interpretarlo y configurar los comandos que ejecutará el dron. Según lo planteado en el trabajo de investigación se implementan dos algoritmos: el clasificador en cascada Haar y la red neuronal convolucional.

Finalmente, se deben realizar pruebas de tiempo de procesamiento y la exactitud en la detección de la mano para ambos algoritmos al variar las condiciones de entorno.

Recomendaciones y trabajos futuros

Se recomienda que la comunicación entre el dron y la computadora se realice utilizando la librería socket de Python que permite implementar el protocolo de comunicación UDP, el cual es el establecido por defecto en el SDK del dron. Esto se debe codificar en un programa en el cual se especifique el IP del dron y el puerto de comunicación por el cual habrá solo transferencia de comandos. Además, se debe codificar un programa para la transmisión de video para tener la capacidad de procesarlas. Considerar, el procesamiento en paralelo; uno para la captura de imágenes y otro, para el procesamiento.

La implementación del clasificador en cascada Haar se debe emplear con el uso de la librería OpenCV y funciones especiales como 'detectMultiScale' que permite leer un archivo que contiene al algoritmo entrenado para la detección de una mano. La imagen recibida por la cámara del dron debe pasar por una etapa de procesamiento que implica la transformación a escala de grises y la aplicación de un filtro gaussiano. También, con la ayuda del método 'findContours' y 'drawContours' se podrá obtener la cantidad de dedos que se encuentran levantados en la mano detectada.

La red neuronal convolucional a utilizar que se recomienda es YOLOv4 pues tiene una precisión promedio de 43.5% y alcanza en promedio 65 FPS. Esto se podrá utilizar al instalar el entorno de trabajo llamado 'darknet' y las dependencias de software CUDA y cuDNN pues permite un procesamiento más rápido de la imagen. Sin embargo, se debe entrenar a la red con imágenes de manos pues por defecto no tiene esta capacidad. Esto se puede implementar con un GPU gratuito disponible en Google Colab.

Bibliografía

- [1] Federal Aviation Administration, "FAA Aerospace Forecast Fiscal Years 2020-2040", USA. Acceso: mayo 5, 2020. URL: https://www.faa.gov/data_research/aviation/aerospace_forecasts/
- [2] CB Insights, "How drones will impact society: From Fighting War to Forecasting Weather, UAVs Change Everything", Acceso: mayo 7, 2020. URL: <https://www.cbinsights.com/research/drone-impact-society-uav/>
- [3] S. Y. Tettegah, Y. E. García, "What is emotional about emotional robotics?" in Emotions, Technology, and Health. USA: Academic Press, pp. 85-103
- [4] Y. Yu, X. Wang, Z. Zhong and Y. Zhang, "ROS-based UAV control using hand gesture recognition," 2017 29th Chinese Control And Decision Conference (CCDC), Chongqing, 2017, pp. 6795-6799, doi: 10.1109/CCDC.2017.7978402.
- [5] A. Sarkar, K. A. Patel, R. K. G. Ram and G. K. Capoor, "Gesture control of drone using a motion controller," 2016 International Conference on Industrial Informatics and Computer Systems (CIICS), Sharjah, 2016, pp. 1-5, doi: 10.1109/ICCSII.2016.7462401.
- [6] K. Natarajan, T. D. Nguyen and M. Mete, "Hand Gesture Controlled Drones: An Open Source Library," 2018 1st International Conference on Data Intelligence and Security (ICDIS), South Padre Island, TX, 2018, pp. 168-175, doi: 10.1109/ICDIS.2018.00035.
- [7] Jessica R. Cauchard, Jane L. E, Kevin Y. Zhai, and James A. Landay. 2015. Drone & me: an exploration into natural human-drone interaction. In Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15). Association for Computing Machinery, New York, NY, USA, 361–365. DOI: <https://doi.org/10.1145/2750858.2805823>
- [8] W. S. Ng and E. Sharlin, "Collocated interaction with flying robots," 2011 RO-MAN, Atlanta, GA, 2011, pp. 143-149, doi: 10.1109/ROMAN.2011.6005280.
- [9] Parrot. "Panel ES – Edito – Parrot AR.Drone 2.0 Elite Edition" parrot.com, <https://www.parrot.com/es/drones/parrot-ardrone-20-elite-edition>. Acceso: mayo 7, 2020.
- [10] Parrot. "Parrot Mambo Fly" parrot.com, <https://www.parrot.com/es/drones/parrot-mambo-fly#>. Acceso: mayo 7, 2020.
- [11] Tello. "Tello Feel the fun" ryzerobotics.com, <https://www.ryzerobotics.com/tello>. Acceso: mayo 7, 2020.
- [12] J.A. Jimenez, "Using Drone Tello Edu for educational purposes," M.S. thesis. Universitat Politècnica de Catalunya, Catalunya, 2019 [Online]. URL: <https://upcommons.upc.edu/bitstream/handle/2117/173834/memoria.pdf?sequence=1&isAllowed=y>
- [13] E. Joy, S. Chandran, C. George, A. A. Sabu and D. Madhu, "Gesture Controlled Video Player A Non-Tangible Approach to Develop a Video Player Based on Human Hand Gestures Using Convolutional Neural Networks," 2018 Second International Conference on

Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 56-60, doi: 10.1109/ICCONS.2018.8662901.

[14] K. Deherkar, G. Martin, N. George and V. Maurya, "Gesture Controlled Virtual Reality Based Conferencing," 2018 International Conference on Smart City and Emerging Technology (ICSCET), Mumbai, 2018, pp. 1-4, doi: 10.1109/ICSCET.2018.8537334.

[15] UltraLeap. UltraLeap Leap Motion Controller. Acceso: mayo 8, 2020. [Online]. URL: https://www.ultraleap.com/datasheets/Leap_Motion_Controller_Datasheet.pdf

[16] ASUS. Xtion PRO Live. Acceso: mayo 8, 2020. [Online]. URL: https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/specifications/

[17] Hired. "2020 State of Software Engineers - Hired", hired.com <https://hired.com/state-of-software-engineers>. Acceso: mayo 8, 2020

[18] Hardware Partnership, "About OpenCV", opencv.org <https://opencv.org/about/>. Acceso: mayo 9, 2020.

[19] Songyan Ma and Lu Bai, "A face detection algorithm based on Adaboost and new Haar-Like feature," 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, 2016, pp. 651-654, doi: 10.1109/ICSESS.2016.7883152.

[20] J. Zhu and Z. Chen, "Real Time Face Detection System Using Adaboost and Haar-like Features," 2015 2nd International Conference on Information Science and Control Engineering, Shanghai, 2015, pp. 404-407, doi: 10.1109/ICISCE.2015.95.

[21] C. Zhang, G. Liu, X. Zhu and H. Cai, "Face Detection Algorithm Based on Improved AdaBoost and New Haar Features," 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Suzhou, China, 2019, pp. 1-5, doi: 10.1109/CISP-BMEI48845.2019.8965841.

[22] I. Dhall, S. Vashisth and G. Aggarwal, "Automated Hand Gesture Recognition using a Deep Convolutional Neural Network model," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2020, pp. 811-816, doi: 10.1109/Confluence47617.2020.9057853.

[23] M. R. Islam, U. K. Mitu, R. A. Bhuiyan and J. Shin, "Hand Gesture Feature Extraction Using Deep Convolutional Neural Network for Recognizing American Sign Language," 2018 4th International Conference on Frontiers of Signal Processing (ICFSP), Poitiers, 2018, pp. 115-119, doi: 10.1109/ICFSP.2018.8552044.

[24] F. Zhan, "Hand Gesture Recognition with Convolution Neural Networks," 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI), Los Angeles, CA, USA, 2019, pp. 295-298, doi: 10.1109/IRI.2019.00054.

[25] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517.

[26] L. Yun and Z. Peng, "An Automatic Hand Gesture Recognition System Based on Viola-Jones Method and SVMs," 2009 Second International Workshop on Computer Science and Engineering, Qingdao, 2009, pp. 72-76, doi: 10.1109/WCSE.2009.769.

[27] Q. Chen, N. D. Georganas and E. M. Petriu, "Real-time Vision-based Hand Gesture Recognition Using Haar-like Features," 2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007, Warsaw, 2007, pp. 1-6, doi: 10.1109/IMTC.2007.379068.

[28] Mohamed Elgendy, "Welcome to Computer Vision" in Deep Learning for Vision Systems, Manning, 2020.

[29] Soret Lee, "Understanding Face Detection with the Viola-Jones Object Detection Framework", Towards data science, 2020. <https://towardsdatascience.com/understanding-face-detection-with-the-viola-jones-object-detection-framework-c55cc2a9da14> Acceso: junio 15, 2020

[30] Valentina Alto, "Understanding AdaBoost for Decision Tree", Towards data science, 2020. <https://towardsdatascience.com/understanding-adaboost-for-decision-tree-ff8f07d2851> Acceso: junio 20, 2020

[31] theAIGuysCode (2020) Forked repository and added conversi3n Python script. https://github.com/theAIGuysCode/OIDv4_ToolKit. Acceso: julio 15, 2020

[32] dtaneja123 (2018) Hand_Recognition. https://github.com/dtaneja123/Hand_Recognition. Acceso: agosto 2, 2020